

Figure - Bug #118644

【IOVDEV-33486】 【内部】 【VC1】 【0072】 【台架】 【EE环境】 【副屏】 副屏死机冻屏恢复后触摸副屏自动重启

2023-06-08 15:31 - CDTs_Test 吴诗雨

Status:	CLOSED	Start date:	2023-06-08
Priority:	Urgent	Due date:	
Assignee:	CDTS_Test 吴诗雨	% Done:	0%
Category:	冻屏	Estimated time:	0.00 hour
Target version:		Found Version:	副屏：01.00.0106.C104
Need_Info:	--	Degrated:	--
Resolution:	DUPLICATE	Verified Version:	
Severity:	Critical	Fixed Version:	
Reproducibility:	Rarely	Root cause:	
Test Type:	ST		
Description			
【环境信息】 EE环境 【问题出现时间】：2023-06-08 10:31 【开机次数】：第275次 【前提条件】： 1.副屏上电 2.下载并安装王者荣耀、火影忍者、原神 3.控制中心打开录屏 【问题操作步骤】： 1.将三个游戏拖拽至同一个文件夹内 2.依次打开游戏返回HOME页 【实际结果】：卡死在打开文件夹界面，恢复后触摸滑动点击屏幕自动重启 【期望结果】：操作设备正常运行，流畅不卡顿 【备注】：副屏无响应是否为内存泄漏，未开启收集log，只是切换游戏，为什么冻屏时录屏依旧进行中，恢复冻屏后触摸屏幕为什么自动重启。早上重启的，下午刚发的收集黑屏死机脚本因此bug没有debuginfo.txt文件 台架实时Log地址：https://ofs.human-horizons.com/#/download/index/yikAOEt67Xs%3D 台架历史Log地址：https://ofs.human-horizons.com/#/download/index/CVzi0kg1KPg%3D 台架视频地址：https://ofs.human-horizons.com/#/download/index/QYxbUHEn3KU%3D 【影响版本】：副屏：01.00.0106.C104 【VIN】：HRYTTESTVINMGM400 【硬件版本号】：/ 【是否能恢复】：			

History

#1 - 2023-06-08 15:32 - CDTs_Test 吴诗雨

日志已上传，视频较大，仍在下载：ftp://cdiot@192.168.87.46/Pre_figure/Test_Log/118644

#2 - 2023-06-08 15:47 - CDTs_Test 吴诗雨

视频已上传

#3 - 2023-06-08 15:57 - CDTs-TEST 周婷

- Category set to 冻屏

#4 - 2023-06-08 19:34 - CD FW 曹覃刚

- Status changed from New to ASSIGNED

- Assignee changed from CD FW 曹覃刚 to CD FW 郑雷

■ 我的分析

发生了下面的anr, block在createTaskSnapshot, 导致重启
这个问题是在 #118646 之后两分钟发生的, 应该有关联

anr_2023-06-08-10-34-14-284

anr_2023-06-08-10-35-14-336

system_server anr

block在下面这个地方,

```
"Binder:1566_10" prio=5 tid=136 Native | group="main" sCount=1 ucsCount=0 flags=1 obj=0x13088560 self=0xb4000075e92719c0 | sysTid=7605 nice=-2 cgrp=default  
sched=0/0 handle=0x73cb0d1cb0 | state=S schedstat=( 5313109889 794988810 9911 ) utm=444 stm=86 core=3 HZ=100 | stack=0x73cafd000-0x73cafdc000  
stackSize=991KB | held mutexes=  
native: #00 pc 00000000000a1c88 /apex/com.android.runtime/lib64/bionic/libc.so (__ioctl+8)  
native: #01 pc 000000000005b8d4 /apex/com.android.runtime/lib64/bionic/libc.so (ioctl+156)  
native: #02 pc 0000000000053070 /system/lib64/libbinder.so (android::IPCThreadState::talkWithDriver(bool)+292)  
native: #03 pc 00000000000542e4 /system/lib64/libbinder.so (android::IPCThreadState::waitForResponse(android::Parcel*, int*)+120)  
native: #04 pc 0000000000053ff0 /system/lib64/libbinder.so (android::IPCThreadState::transact(int, unsigned int, android::Parcel const&, android::Parcel*, unsigned  
int)+224)  
native: #05 pc 000000000004bafc /system/lib64/libbinder.so (android::BpBinder::transact(unsigned int, android::Parcel const&, android::Parcel*, unsigned int)+196)  
native: #06 pc 00000000000c73a0 /system/lib64/libgui.so (android::BpSurfaceComposer::captureLayers(android::LayerCaptureArgs const&,  
android::sp<android::gui::IScreenCaptureListener> const&)+476)  
native: #07 pc 00000000000f73f8 /system/lib64/libgui.so (android::ScreenshotClient::captureLayers(android::LayerCaptureArgs const&,  
android::sp<android::gui::IScreenCaptureListener> const&)+80)  
native: #08 pc 00000000001322d4 /system/lib64/libandroid_runtime.so (android::nativeCaptureLayers(_JNIEnv*, _jclass*, _jobject*, _jobject*)+740)  
at android.view.SurfaceControl.nativeCaptureLayers(Native method)  
at android.view.SurfaceControl.captureLayers(SurfaceControl.java:2403)  
at android.view.SurfaceControl.captureLayers(SurfaceControl.java:2371)  
at android.view.SurfaceControl.captureLayersExcluding(SurfaceControl.java:2393)  
at com.android.server.wm.TaskSnapshotController.createTaskSnapshot(TaskSnapshotController.java:453)  
at com.android.server.wm.TaskSnapshotController.createTaskSnapshot(TaskSnapshotController.java:363)  
at com.android.server.wm.TaskSnapshotController.snapshotTask(TaskSnapshotController.java:483)  
at com.android.server.wm.TaskSnapshotController.snapshotTask(TaskSnapshotController.java:470)  
at com.android.server.wm.TaskSnapshotController.snapshotTasks(TaskSnapshotController.java:193)  
at com.android.server.wm.TaskSnapshotController.snapshotTasks(TaskSnapshotController.java:175)  
at com.android.server.wm.RecentsAnimationController$2.screenshotTask(RecentsAnimationController.java:220)  
- locked <0x084cfd4> (a com.android.server.wm.WindowManagerGlobalLock)  
at android.view.IRecentsAnimationController$Stub.onTransact(IRecentsAnimationController.java:273)  
at android.os.Binder.execTransactInternal(Binder.java:1179)  
at android.os.Binder.execTransact(Binder.java:1143)
```

06-08 10:34:31.784 1566 1671 W Watchdog: *** WATCHDOG KILLING SYSTEM PROCESS: Blocked in monitor com.android.server.am.ActivityManagerService on
foreground thread (android.fg), Blocked in handler on display thread (android.display), Blocked in handler on ActivityManager (ActivityManager)

//system_server无响应被杀重启

■ 下一步计划

- 1.结合 #118646 一起分析
- 2.检查createTaskSnapshot block的原因

#5 - 2023-06-09 19:02 - CD FW 郑雷

[当前分析]

- 1.从视频从能看到王者荣耀、火影忍者、原神等app打开失败
- 2.冻屏是因为出现WATCHDOG，并且系统重启。

分析log：

- 1.关于王者荣耀、火影忍者、原神 app打开失败的log：

06-08 10:33:03.766 681 681 D Zygote : Forked child process 13504

06-08 10:33:03.766 1566 1677 I am_proc_start:

[0,13504,10096,com.tencent.KiHan,pre-top-activity,{com.tencent.KiHan/com.tencent.gcloud.msdk.core.policy.MSDKPolicyActivity}]

06-08 10:33:03.767 1566 1677 I ActivityManager: Start proc 13504:com.tencent.KiHan/u0a96 for pre-top-activity

{com.tencent.KiHan/com.tencent.gcloud.msdk.core.policy.MSDKPolicyActivity}

.....

06-08 10:33:03.931 845 845 W SDM : HWCDisplay::GetHWCLayer: [0] GetLayer(612) failed: no such layer

06-08 10:33:03.935 1002 1002 W HwcComposer: command 0x3000000 generated error 3

06-08 10:33:04.535 1890 2590 D CCodecBufferChannel: [c2.android.aac.encoder#836] DEBUG: elapsed: n=4 [in=0 pipeline=0 out=0]

06-08 10:33:04.535 1890 2590 D PipelineWatcher: DEBUG: elapsed 0 / 4

.....

06-08 10:33:08.950 1566 1676 W ActivityManager: Process ProcessRecord{e1c0c39 13475:com.tencent.tmgp.sgame/u0a112} failed to attach

06-08 10:33:08.950 1566 1676 I am_process_start_timeout: [0,13475,10112,com.tencent.tmgp.sgame]

06-08 10:33:08.950 1566 1676 I am_uid_stopped: 10112

06-08 10:33:08.951 1566 1676 I ActivityManager: Killing 13475:com.tencent.tmgp.sgame/u0a112 (adj -10000): start timeout

06-08 10:33:08.951 1566 1676 I am_kill : [0,13475,com.tencent.tmgp.sgame,-10000,start timeout]

当启动目标Activity前，会先pause当前正显示的Activity，之后才显示目标Activity。在startPausingLocked()方法中会获取LaunchWakelock，并设置超时监听

```
final boolean startPausingLocked(boolean userLeaving, boolean uiSleeping,
```

```
    ActivityRecord resuming, String reason) {
```

```
    .....
```

```
    // If we are not going to sleep, we want to ensure the device is
```

```
    // awake until the next activity is started.
```

```
    if (!uiSleeping && !mAtmService.isSleepingOrShuttingDownLocked()) {
```

```
        mTaskSupervisor.acquireLaunchWakelock();
```

```
    }
```

```
    .....
```

```
}
```

通过mHandler发送延迟消息的方式来进行超时设置，what为LAUNCH_TIMEOUT_MSG，LAUNCH_TIMEOUT为10s

```
void acquireLaunchWakelock() {
```

```
    if (VALIDATE_WAKE_LOCK_CALLER && Binder.getCallingUid() != Process.myUid()) {
```

```
        throw new IllegalStateException("Calling must be system uid");
```

```
    }
```

```
    mLaunchingActivityWakeLock.acquire();
```

```
    if (!mHandler.hasMessages(LAUNCH_TIMEOUT_MSG)) {
```

```
        // To be safe, don't allow the wake lock to be held for too long.
```

```
        mHandler.sendMessageDelayed(LAUNCH_TIMEOUT_MSG, LAUNCH_TIMEOUT);
```

```
    }
```

```
}
```

在ActivityStackSupervisorHandler的handleMessage方法的LAUNCH_TIMEOUT_MSG消息

```
public void handleMessage(Message msg) {
    switch (msg.what) {
        case LAUNCH_TIMEOUT_MSG: {
            if (mLaunchingActivityWakeLock.isHeld()) {
                Slog.w(TAG, "Launch timeout has expired, giving up wake lock!");
                if (VALIDATE_WAKE_LOCK_CALLER
                    && Binder.getCallingUid() != Process.myUid()) {
                    throw new IllegalStateException("Calling must be system uid");
                }
                //释放锁
                mLaunchingActivityWakeLock.release();
            }
            break;
        }
        // . . .
    }
}
```

在目标Activity启动完成后，AMS会移除LAUNCH_TIMEOUT_MSG消息。

com.tencent.KiHan 启动失败的原因是Activity启动超时，AMS kill 了进程。

导致Activity启动失败的原因，从log分析有如下异常，显示composer-service （845） 和 surfaceflinger （1002） 出现了异常。导致view渲染失败，activity启动失败。

```
06-08 10:33:03.931 845 845 W SDM : HWCDisplay::GetHWCLayer: [0] GetLayer(612) failed: no such layer
```

```
06-08 10:33:03.935 1002 1002 W HwcComposer: command 0x3000000 generated error 3
```

2. 关于WATCHDOG Exception

```
06-08 10:34:31.784 1566 1671 W Watchdog: * WATCHDOG KILLING SYSTEM PROCESS: Blocked in monitor com.android.server.am.ActivityManagerService on foreground thread (android.fg), Blocked in handler on display thread (android.display), Blocked in handler on ActivityManager (ActivityManager)
```

```
06-08 10:34:31.784 1566 1671 W Watchdog: android.fg annotated stack trace:
```

```
06-08 10:34:31.784 1566 1671 W Watchdog: at android.os.Process.setThreadPriority(Native Method)
```

```
06-08 10:34:31.784 1566 1671 W Watchdog: at com.android.server.ThreadPriorityBooster.boost(ThreadPriorityBooster.java:50)
```

```
06-08 10:34:31.784 1566 1671 W Watchdog: at com.android.server.am.ActivityManagerService.boostPriorityForLockedSection(ActivityManagerService.java:733)
```

```
06-08 10:34:31.784 1566 1671 W Watchdog: at com.android.server.am.ActivityManagerService.broadcastIntentWithFeature(ActivityManagerService.java:13720)
```

```
06-08 10:34:31.784 1566 1671 W Watchdog: - locked <0x0996d039> (a com.android.server.am.ActivityManagerService)
```

```
06-08 10:34:31.785 1566 1671 W Watchdog: at android.app.ContextImpl.sendBroadcastAsUser(ContextImpl.java:1431)
```

```
06-08 10:34:31.785 1566 1671 W Watchdog: at android.app.ContextImpl.sendBroadcastAsUser(ContextImpl.java:1402)
```

```
06-08 10:34:31.785 1566 1671 W Watchdog: at
```

```
com.android.server.DropBoxManagerService$DropBoxManagerBroadcastHandler.prepareAndSendBroadcast(DropBoxManagerService.java:284)
```

```
06-08 10:34:31.785 1566 1671 W Watchdog: at
```

```
com.android.server.DropBoxManagerService$DropBoxManagerBroadcastHandler.handleMessage(DropBoxManagerService.java:266)
```

```
06-08 10:34:31.785 1566 1671 W Watchdog: at android.os.Handler.dispatchMessage(Handler.java:106)
```

```
06-08 10:34:31.785 1566 1671 W Watchdog: at android.os.Looper.loopOnce(Looper.java:201)
```

```
06-08 10:34:31.785 1566 1671 W Watchdog: at android.os.Looper.loop(Looper.java:288)
```

```
06-08 10:34:31.785 1566 1671 W Watchdog: at android.os.HandlerThread.run(HandlerThread.java:67)
```

```
06-08 10:34:31.785 1566 1671 W Watchdog: at com.android.server.ServiceThread.run(ServiceThread.java:44)
```

```
06-08 10:34:31.785 1566 1671 W Watchdog: android.display annotated stack trace:
```

```
06-08 10:34:31.785 1566 1671 W Watchdog: at android.os.MessageQueue.nativePollOnce(Native Method)
```

```
06-08 10:34:31.785 1566 1671 W Watchdog: at android.os.MessageQueue.next(MessageQueue.java:335)
```

06-08 10:34:31.785 1566 1671 W Watchdog: at android.os.Looper.loopOnce(Looper.java:161)
06-08 10:34:31.785 1566 1671 W Watchdog: at android.os.Looper.loop(Looper.java:288)
06-08 10:34:31.785 1566 1671 W Watchdog: at android.os.HandlerThread.run(HandlerThread.java:67)
06-08 10:34:31.785 1566 1671 W Watchdog: at com.android.server.ServiceThread.run(ServiceThread.java:44)
06-08 10:34:31.785 1566 1671 W Watchdog: ActivityManager annotated stack trace:
06-08 10:34:31.785 1566 1671 W Watchdog: at android.os.MessageQueue.nativePollOnce(Native Method)
06-08 10:34:31.785 1566 1671 W Watchdog: at android.os.MessageQueue.next(MessageQueue.java:335)
06-08 10:34:31.785 1566 1671 W Watchdog: at android.os.Looper.loopOnce(Looper.java:161)
06-08 10:34:31.785 1566 1671 W Watchdog: at android.os.Looper.loop(Looper.java:288)
06-08 10:34:31.785 1566 1671 W Watchdog: at android.os.HandlerThread.run(HandlerThread.java:67)
06-08 10:34:31.785 1566 1671 W Watchdog: at com.android.server.ServiceThread.run(ServiceThread.java:44)
06-08 10:34:31.785 1566 1671 W Watchdog: * GOODBYE!
06-08 10:34:31.785 1566 1671 I Process : Sending signal. PID: 1566 SIG: 9

"android.fg" prio=5 tid=11 Blocked | group="main" sCount=1 ucsCount=0 flags=1 obj=0x13080510 self=0xb4000075e9240f00 | sysTid=1665 nice=0 cgrp=default
sched=0/0 handle=0x73ecc80cb0 | state=S schedstat=(610080091 682256943 2882) utm=40 stm=20 core=0 HZ=100 | stack=0x73ecb7d000-0x73ecb7f000
stackSize=1039KB | held mutexes=
at com.android.server.am.ActivityManagerService.monitor(ActivityManagerService.java:15167)
- waiting to lock <0x0996d039> (a com.android.server.am.ActivityManagerService) held by thread 21
at com.android.server.Watchdog\$HandlerChecker.run(Watchdog.java:275)
at android.os.Handler.handleCallback(Handler.java:938)
at android.os.Handler.dispatchMessage(Handler.java:99)
at android.os.Looper.loopOnce(Looper.java:201)
at android.os.Looper.loop(Looper.java:288)
at android.os.HandlerThread.run(HandlerThread.java:67)
at com.android.server.ServiceThread.run(ServiceThread.java:44)

"ActivityManager" prio=5 tid=21 Blocked | group="main" sCount=1 ucsCount=0 flags=1 obj=0x13080bb0 self=0xb4000075e9252520 | sysTid=1676 nice=-2 cgrp=default
sched=0/0 handle=0x73e212acb0 | state=S schedstat=(1168414253 570734027 4220) utm=46 stm=70 core=0 HZ=100 | stack=0x73e2027000-0x73e2029000
stackSize=1039KB | held mutexes=
at com.android.server.wm.ActivityTaskManagerService\$LocalService.onProcessUnMapped(ActivityTaskManagerService.java:5584)
- waiting to lock <0x084cfd4> (a com.android.server.wm.WindowManagerGlobalLock) held by thread 136
at com.android.server.am.ActivityManagerService.removePidIfNoThreadLocked(ActivityManagerService.java:960)
at com.android.server.am.ActivityManagerService.processStartTimedOutLocked(ActivityManagerService.java:4312)
at com.android.server.am.ActivityManagerService.access\$100(ActivityManagerService.java:436)
at com.android.server.am.ActivityManagerService\$MainHandler.handleMessage(ActivityManagerService.java:1701)
- locked <0x0996d039> (a com.android.server.am.ActivityManagerService)
at android.os.Handler.dispatchMessage(Handler.java:106)
at android.os.Looper.loopOnce(Looper.java:201)
at android.os.Looper.loop(Looper.java:288)
at android.os.HandlerThread.run(HandlerThread.java:67)
at com.android.server.ServiceThread.run(ServiceThread.java:44)

"Binder:1566_10" prio=5 tid=136 Native | group="main" sCount=1 ucsCount=0 flags=1 obj=0x13088560 self=0xb4000075e92719c0 | sysTid=7605 nice=-2 cgrp=default
sched=0/0 handle=0x73cb0d1cb0 | state=S schedstat=(5313235566 794988810 9913) utm=444 stm=86 core=3 HZ=100 | stack=0x73cafd000-0x73cafdc000
stackSize=991KB | held mutexes=
native: #00 pc 00000000000a1c88 /apex/com.android.runtime/lib64/bionic/libc.so (__ioctl+8)
native: #01 pc 000000000005b8d4 /apex/com.android.runtime/lib64/bionic/libc.so (ioctl+156)
native: #02 pc 0000000000053070 /system/lib64/libbinder.so (android::IPCThreadState::talkWithDriver(bool)+292)
native: #03 pc 00000000000542e4 /system/lib64/libbinder.so (android::IPCThreadState::waitForResponse(android::Parcel*, int*)+120)
native: #04 pc 0000000000053ff0 /system/lib64/libbinder.so (android::IPCThreadState::transact(int, unsigned int, android::Parcel const&, android::Parcel*, unsigned
int)+224)
native: #05 pc 000000000004bafc /system/lib64/libbinder.so (android::BpBinder::transact(unsigned int, android::Parcel const&, android::Parcel*, unsigned int)+196)

```
native: #06 pc 0000000000c73a0 /system/lib64/libgui.so (android::BpSurfaceComposer::captureLayers(android::LayerCaptureArgs const&,
android::sp<android::gui::IScreenCaptureListener> const&)+476)
native: #07 pc 0000000000f73f8 /system/lib64/libgui.so (android::ScreenshotClient::captureLayers(android::LayerCaptureArgs const&,
android::sp<android::gui::IScreenCaptureListener> const&)+80)
native: #08 pc 0000000001322d4 /system/lib64/libandroid_runtime.so (android::nativeCaptureLayers(_JNIEnv*, _jclass*, _jobject*, _jobject*)+740)
at android.view.SurfaceControl.nativeCaptureLayers(Native method)
at android.view.SurfaceControl.captureLayers(SurfaceControl.java:2403)
at android.view.SurfaceControl.captureLayers(SurfaceControl.java:2371)
at android.view.SurfaceControl.captureLayersExcluding(SurfaceControl.java:2393)
at com.android.server.wm.TaskSnapshotController.createTaskSnapshot(TaskSnapshotController.java:453)
at com.android.server.wm.TaskSnapshotController.createTaskSnapshot(TaskSnapshotController.java:363)
at com.android.server.wm.TaskSnapshotController.snapshotTask(TaskSnapshotController.java:483)
at com.android.server.wm.TaskSnapshotController.snapshotTask(TaskSnapshotController.java:470)
at com.android.server.wm.TaskSnapshotController.snapshotTasks(TaskSnapshotController.java:193)
at com.android.server.wm.TaskSnapshotController.snapshotTasks(TaskSnapshotController.java:175)
at com.android.server.wm.RecentsAnimationController$2.screenshotTask(RecentsAnimationController.java:220)
- locked <0x084cfd4> (a com.android.server.wm.WindowManagerGlobalLock)
at android.view.IRecentsAnimationController$Stub.onTransact(IRecentsAnimationController.java:273)
at android.os.Binder.execTransactInternal(Binder.java:1179)
at android.os.Binder.execTransact(Binder.java:1143)
```

system_server的Binder:1566_10线程在调用createTaskSnapshot时会调用nativeCaptureLayers方法，nativeCaptureLayers方法被binder对端block住。而android::BpSurfaceComposer::captureLayers的对端是SurfaceFlinger进程

[下一步Action]

检查"surfaceflinger"进程异常的原因

#6 - 2023-06-12 17:12 - CD FW 郑雷

[当前分析]

从目前WATCHDOG的信息看，是被block在SurfaceFlinger进程，但是从SurfaceFlinger的堆栈看，看不到被block的原因。

在WATCHDOG异常 system_server重启之前添加doSysRq('c')抓取ramdump信息。

<https://dev.thundercomm.com/gerrit/c/general/platform/frameworks/base/+205852>

备注：

这个WATCHDOG的信息和<https://share.thundersoft.com/redmine/issues/118738>很像，原因有可能也像118738，SurfaceFlinger被底层block了。

[下一步Action]

编译抓取ramdump的版本，复现问题，查看被SurfaceFlinger block的原因。

#7 - 2023-06-15 21:03 - CD FW 曹覃刚

- Status changed from ASSIGNED to RESOLVED

- Assignee changed from CD FW 郑雷 to CDTS_Test 吴诗雨

- Resolution changed from -- to DUPLICATE

转JIRA跟踪

#8 - 2023-07-10 17:16 - CDTS_Test 吴诗雨

- Status changed from *RESOLVED* to *VERIFIED*

#9 - 2023-07-10 17:16 - CDTS_Test 吴诗雨

- Status changed from *VERIFIED* to *CLOSED*